http://www.pptph.com.cn

Delphi 编程疑难详解

季雪岗　王晓辉　张宏林　等编著

高手指路系列丛书

人民邮电出版社

高手指路系列丛书

Delphi 编程疑难详解

季雪岗　王晓辉　张宏林　等编著

人民邮电出版社

内容提要

Delphi 是当前最强大、最灵活、最快速的应用程序开发工具之一，它将可视化界面与面向对象的 Pascal 语言完美地结合在一起，为程序员开辟了崭新的编程天地。本书针对程序员编程中经常碰到的疑难问题，通过实例，详细介绍了编程方法以及相关技巧。

本书阐述了五大部分方面的疑难问题：基础编程、系统编程、多媒体编程、数据库编程、辅助制作等。

本书技术内容新颖、阐述方法明晰、覆盖范围较广、实用性强，适合于那些从事 Delphi 开发编程、有一定经验而现在又需要进一步提高技术水平的程序员或相关技术人员阅读。对于 Delphi 的初学者来说，本书也可以作为了解编程过程中常见问题解决方法的一个集锦，书中提供的代码或方法可解燃眉之急。

高手指路系列丛书

Delphi 编程疑难详解

◆ 编　　著　季雪岗　王晓辉　张宏林　等

责任编辑　张立科

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街 14 号

邮编　100061　　电子函件　315@pptph.com.cn

网址　http://www.pptph.com.cn

北京汉魂图文设计有限公司制作

北京鸿佳印刷厂印刷

新华书店总店北京发行所经销

◆ 开本：787×1092　1/16

印张：30.5

字数：768 千字　　2000 年 7 月第 1 版

印数：1－5 000 册　　2000 年 7 月北京第 1 次印刷

ISBN 7-115-08635-4/TP·1712

定价：45.00 元

前言

Delphi 从诞生的第一天起，就获得了不少的殊荣。*PC Magazine* 在开发工具一栏中给 Delphi 贴上了技术出众的标签。*Computer World* 报告说 Delphi 在顾客的满意度记分牌上得分最高。甚至连支持 Delphi 竞争者的出版商也高度评价这一产品，Delphi 在《Visual Basic 编辑者杂志》上也荣获编辑推荐奖。

作为一个老程序员，已经从“DOS 时代”走到了现在的“Windows 时代”，开发工具也经历了同样的“变革”。记得在 DOS 年代中，Borland 的 Turbo 系列（Turbo C、Turbo Pascal 等）开发工具的集成开发环境就受到过程序员们的青睐和赞许；在 Windows 时代，Borland 公司（现在为 Inprise 公司，但作者还是习惯叫 Borland 公司）又将它的这一优点在 Windows 中发挥得淋漓尽致，代表作之一就是 Delphi。从 1994 年诞生起，它就成了 Delphi 程序员心中的“女神”。可以说，Borland 公司在推动开发工具的发展过程中起了重要的作用。

目前，Delphi 已经成为作者生活中的一部分了。作者经常在网上与同行们一起探讨编程中所碰到的问题、收集和下载控件、发布控件、整理 Delphi 一些资料、上网下载 Delphi 的补丁和升级包等。随着 Delphi 的不断发展和升级，作者从 Delphi 1.0 起一直使用到现在的 Delphi 5.0，其中有桌面版(Desktop)、专业版(Professional)、客户\服务器版(Client\Server)以及企业版(Enterprise)等。作者用每一个版本开发过一些项目或者课题。在开发过程中与很多程序员一样，遇到了各种各样的问题，但同时也积累了很多问题的解决方法。作为老的 Delphi 程序员，有责任将 Delphi 推荐给大家，更有责任将自己碰到过的问题的解决方法告诉同行们，与 Delphi 程序员共同分享这一份果实。

本书所选的内容主要是程序员经常碰到的疑难问题，例如，在多媒体中如何控制光驱、如何实现视屏捕捉、如何制作屏幕保护程序等，例子都是一些切实可用的程序。在内容安排上，尽量做到由浅入深，基本上是按照问题的提出、问题的解决和问题的总结三部曲的格式来进行阐述的。

本书主要分五部分：基础编程；系统编程；多媒体编程；数据库编程；辅助制作等。

书中，第一部分为基础编程，主要阐述了 Pascal 语法新特点以及 VCL 编程中常见的疑难问题。VCL 编程中主要讲述了窗体设计、MDI 应用、菜单、控件等有关问题。

第二部分为系统编程，主要阐述了资源文件的使用、文件与驱动器的控制以及使用、剪贴板的利用、串口通信的实现、打印机的控制、Delphi 的消息系统机理、Windows 的系统控制、API 调用等疑难问题。

第三部分为多媒体编程，主要阐述了图形、图像以及多媒体实现中常见的疑难问题。

第四部分为数据库编程，主要阐述了 dbgrid 高级应用、快速报表的制作、数据库引擎的编程与使用、数据库控件的使用以及 MIDAS 技术的机理等问题。

第五部分为辅助制作，除了编程，对于一个完整的系统还需要制作帮助文件以及安装文件，该部分主要阐述如何制作应用系统的帮助文件以及安装程序等问题。

本书主要由季雪岗、王晓辉、张宏林编写。另外，为本书编写提供帮助的人员有：李新友、彭晓东、王德英、田苗、岳庆生、赵子忠、曹冬炎、屈山、吴海、董怡、张雷、王湘云、彭志强、李伟、苗立东、徐顺兴、徐扬、王明磊、刘新、魏东、鲁波、赵源、赵烨、赵彦、钟小刚、刘芳、刘慎锋、余淼、黄明、曹伟齐、吴奎元等。

由于作者水平有限，书中缺点与错误在所难免，我们诚恳希望读者批评指正。

作者

2000 年 5 月

E-Mail:jixg@thunis.com

目录

第一章

Object Pascal 语法新特点

热点透视

Delphi4 和 Delphi5 中对 Object Pascal 语言本身也做了一些改进，增加了对动态数组、函数重载、默认参数、64 位整数、32 位无符号整数、改变了的实数类型以及 Implements 接口等支持。由于动态数组、函数重载以及缺省参数的使用对程序员的编程方式、风格甚至算法等都会产生影响，程序员实现一些程序变得更加容易，并且使程序的可读性更强，因此，本章将针对这些方面进行介绍。另外，还介绍了如何动态创建对象。

实现动态数组

遇到难题

相信不少编程人员对 Pascal 中的数组已经相当熟悉了，但在 Delphi3 之前，有的用户还也许没有使用过动态数组，尽管在 Pascal 中同 C 一样可以通过指针来使用动态数据。以前，为了实现动态数组，通常由指针变量来定义一条链，这样操作起来非常繁琐，同时还得让每一个元素必须记录前后元素的地址。现在，在 Delphi4 和 Delphi5 中已经增加了动态数组这一数据类型。那么，在新版的 Delphi 中又如何使用动态数组呢？

钥匙在此

跟使用静态数组相类似，为了使用动态数组，必须经过变量的声明和变量的使用这两个过程。但是在使用之前必须指定数组的大小以及给数组分配空间。

1. 动态数组声明

在 Delphi4 或 Delphi5 中，可以声明静态数组，其声明格式如下：

```
A: array[1..100] of string;
```

现在还可以声明动态数组。动态数组的维数和长度在编译期间是不确定的。动态数组只指定类型信息（如数组的维数以及元素的类型）而不指定元素数。

动态数组并没有固定的长度或大小。不过，动态数组的内存分配会在给它赋值时或将其作为参数传给过程函数 SetLength 时进行分配。动态数组的类型的声明格式为：

```
array of baseType
```

例如，

```
var B: array of string;
```

2. 为动态数组指定大小并分配内存空间

上例中声明了一维字符串动态数组 B，但还没有为 B 分配内存。为了给数组分配内存空间，调用 SetLength 过程函数。例如，在上述声明基础上，调用

```
SetLength(B, 20);
```

分配了包含 20 个字符串类型元素的数组，下标索引为 0~19。分配了内存后，就可以像访问一般的数组那样访问动态数组的每一个元素了：

```
B[0]:='我是一个字符串！';
showmessage(B[0]);
```

动态数组是以整数作为下标的，而且总是从 0 开始。

动态数组也可以是多维的。若要指定多维动态数组，可以参考下面的代码：

```
var
    //基类型为整数的二维动态数组
    IA:array of array of integer;
```

要为多维的动态数组分配内存，可以参考下面的代码：

```
    //IA 将是 5X5 的整型数组
    SetLength(IA,5,5);
```

访问多维的动态数组就像访问一般的多维数组一样，如：

```
    IA[0,3]:=2;
```

在一些函数或过程声明中，数组参数是以 array of baseType 声明的，没有指定任何下标索引，例如：

```
function CheckStrings(A: array of string): Boolean;
```

这就意味着该函数主要关心数组的基类型，而不是数组的大小、索引方法或者是动态还是静态的方式分配内存等方面内容。

3. 为动态数组释放空间

动态数组具有生存期自管理功能。因此，没有必要专门去释放它所占有的内存。不过，如果想在动态数组超出其作用域之前删除它（如果它占用了很多内存的话），则只要用 Nil 对动态数组赋值即可。程序示例如下：

```
B:=Nil;//释放数组空间
```

4. 动态数组相互赋值

与字符串类似，动态数组相互赋值时，赋值的仅仅是对数据的引用，而非数据本身，这一点与一般的数组不同。请看下面的代码：

```
var
  A, B: array of Integer;
begin
  SetLength(A, 4);//为数组 A 分配空间
  A[0] := 1;//为元素 0 赋值
  B := A;
  B[0] := 2;
end;
```

A[0]现在等于几呢？正确答案为 2。这是因为赋值语句 B:=A 并没有为 B 单独分配内存，而是使 B 与 A 引用同一数组。这样，任何对 B 的修改必然会影响到 A。如果想把 A 的数据复制一个副本给 B，就应当调用标准过程 Copy()：

```
B:=Copy(A);
```

上面这行代码执行后，B 与 A 就成为两个独立的数组，尽管它们的初始值是相同的，但是修改其中一个数组的数据不会影响另一个数组。另外，调用 Copy()时，还可以指定从

第几个元素开始复制、复制几个元素等，示例如下：

```
//从第 1 个元素开始复制 2 个元素
B:=Copy(A,1,2);
```

下面来看一下创建多维动态数组的完整例子。

跟我来

为了声明多维动态数组，可以重复使用 array of ...结构，如：

```
type TMessageGrid = array of array of string;
var Msgs: TMessageGrid;
```

这样就声明了一个二维字符串动态数组。如果 I.J 是两个整数型变量，则

```
SetLength(Msgs,I,J);
```

表示分配了大小为 I×J 的数组，并且 Msgs[0,0]表示该数组的第 1 个元素。

另外，也可以使用另一种方式来为动态数组分配内存。如：

```
var Ints: array of array of Integer;
SetLength(Ints,10);
```

可以为 Ints 分配 10 行空间。然后，还可以单独为每一行分配空间。例如：

```
SetLength(Ints[2], 5);
```

为第 3 行分配了 5 个整数空间。此时，还可以给该行中的元素进行赋值，虽然其他的行还未分配列。如 Ints[2,4] := 63。

下面的例子使用了动态数组来创建三角矩阵：

```
Var
  A : array of array of string;
  I, J : Integer;
  s:string;
Begin
  //为数组 A 分配 10 行空间
  SetLength(A, 10);
  //I 变量从 A 的最小下标 0 到 A 的最大下标 9 进行循环
  for I := Low(A) to High(A) do
  begin
    //为 A[I]行分配 I 列空间
    SetLength(A[I], I);
    //J 变量从 A[I]的最小下标到 A[I]的最大下标进行循环
    for J := Low(A[I]) to High(A[I]) do begin
      A[I,J] := IntToStr(I) + ',' + IntToStr(J) + ' ';
      s:=s+A[I,j];
        end;
        s:=s+#13;
```

```
  end;
  showmessage(s);
end;
```

运行该段程序，结果如图 1-1 所示。

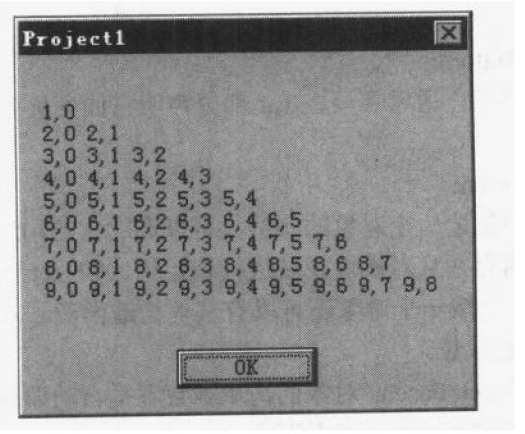
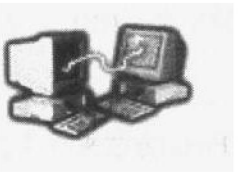


图 1-1　三角矩阵

Low()和 High()两个函数分别返回一个范围内的最小值和最大值。如果该函数的参数为数组的话，就将返回该数组的下标最小值和下标最大值。



动态创建对象

遇到难题

在使用 Delphi 编程时，一般在设计阶段就把整个界面都制作完毕，但有时也会碰到在 Delphi 的应用程序运行时，需要创建一些界面元素的情况，如在某些条件下，需要动态地为窗体创建一个按钮，一个文本框，甚至需要动态地创建一些窗体等。编一个不同风格的界面时更是如此。那么，在程序执行时如何动态地创建这些组件或者其他对象呢？

钥匙在此

在 OOP（面向对象编程）技术中，对象是同时包含数据和操作的实体，它是类的一个实例（OOP 是一项非常成熟的技术，并且有很多资料都详细介绍过这方面的内容，因此，本书将不再赘述）。Delphi 的对象具有面向对象编程所具有的强大优势，全面支持继承、封

装和多态。

1. 声明和创建实例

在使用一个对象之前，首先必须使用关键字 class 来声明该对象的类。对象应当在程序单元文件的 type 部分进行声明，如：

```
type
TmyButton=class(TButton);
```

声明了一种对象类型后，通常还要在 var 部分声明它的变量或实例：

```
var
  myButton:TmyButton;
```

在 Object Pascal 中要创建一个对象的实例，可以调用它的构造函数。构造函数的作用是创建对象的实例、分配内存并且对属性或域进行初始化。Object Pascal 的对象至少要有一个叫 Create()的构造函数，当然一个对象还可以有其他构造函数。对于不同类型的对象来说，Create()可以带不同数量的参数。

与 C++不同的是 Object Pascal 对象的构造函数不是自动调用的，程序员必须自己调用构造函数。调用构造函数的文法可以参考下面的例子：

```
myButton:=TmyButton.Create(Nil);
```

调用构造函数的文法有点特殊。Create()是由类型调用的而不是实例调用的，而其他方法一般只能由实例来调用。通过调用构造函数来创建对象的实例，这就是所谓的实例化。

通过调用构造函数创建了对象的实例后，编译器保证对象的每一个域或属性都已经初始化。这时就可以放心地认为所有的数字都初始化为 0，所有的指针都初始化为 Nil，所有的字符串都初始化为空字符串。

2. 析构函数

当对象使用完毕时，应当及时地调用它的 Free()方法来删除它。Free()方法首先检查对象实例是否为 Nil，如果不是的话，就调用对象的析构函数即 Destroy()。析构与构造的作用恰好相反，它释放先前分配的内存，并做一些清除工作，以保证对象从内存中彻底清除。调用析构的文法请参考下面的例子：

```
myButton.Free;
```

与调用 Create()不同的是，必须通过对象实例来调用 Free()。记住，不要直接调用 Destroy()，而要调用 Free()。

这些方法是怎样放在一个小小的对象中的呢？实际上，这些方法都来自于 Object Pascal 的 Tobject 对象，在 Object Pascal 语言中，所有的对象都是从 Tobject 继承下来的，不管它们在声明的时候有没有列出 Tobject。例如：

```
Type  TjxgObject=Class;
```

相当于

```
Type TjxgObject=class(TObject);
```

下面看一下在窗体中动态创建按钮的例子。

跟我来

新建一个 Delphi 应用程序，清单 1-1 列出了完整的单元。其中，设置窗体名为 FrmMain，相应的单元文件名为 Main.pas，并在 FrmMain 中添加了一个 MyOnClick()过程，同时在窗体的 OnShow 和 OnClose 事件中添加了代码。

清单 1-1　动态创建按钮

```
unit Main;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls;

type
  TFrmMain = class(TForm)
    procedure FormShow(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
    procedure   MyOnClick(Sender: TObject);
  end;

var
  FrmMain: TFrmMain;

implementation

{$R *.DFM}
var
   //声明对象实例
   myButton:TButton;

//自定义过程
procedure   TFrmmain.MyOnClick (Sender: TObject);
begin
      showmessage('我是被动态创建的！');
```

```
end;

procedure TFrmMain.FormShow(Sender: TObject);
begin
    //创建按钮实例
    myButton:=TButton.Create(self);
   //设置动态按钮的属性
    mybutton.left:=20;
    mybutton.Top:=20;
    myButton.Width:=120;
    myButton.Caption:='动态创建的按钮';
    myButton.Parent:=Self;//该语句使按钮在窗体上可见
    //将 MyOnClick 过程赋值给 OnClick 属性（事件）
    myButton.OnClick:=MyOnClick;

end;

procedure TFrmMain.FormClose(Sender: TObject; var Action: TCloseAction);
begin
     //释放实例
      myButton.Free;
end;

end.
```

运行该程序代码，将出现如图 1-2 所示的窗体，即在设计状态为空白的窗体上自动添加了一个按钮，其标题为“动态创建的按钮”，当单击该按钮时将会弹出一个“我是被动态创建的!”对话框，因为已经将该按钮的单击事件赋给了自定义的一个过程 MyOnClick，所以单击鼠标时自动激发了 MyOnClick 过程。

图 1-2　动态创建按钮

关于赋值给对象事件属性的过程或函数的定义，应当符合该事件属性的过程或函数的声明。如，按钮的 OnClick 事件的类型为：

```
type TNotifyEvent = procedure (Sender: TObject) of object;
```